

Fast porous visco-hyperelastic soft tissue model for surgery simulation: application to liver surgery

Stéphanie Marchesseau^a, Tobias Heimann^a, Simon Chatelin^b, Rémy Willinger^b, Hervé Delingette^a

^aAsclepios Research Project, INRIA Sophia Antipolis, France

^bUniversity of Strasbourg, IMFS-CNRS, Strasbourg, France

Abstract

Understanding and modeling liver biomechanics represents a significant challenge due to its complex nature. In this paper, we tackle this issue in the context of real time surgery simulation where a compromise between biomechanical accuracy and computational efficiency must be found. We describe a realistic liver model including hyperelasticity, porosity and viscosity that is implemented within an implicit time integration scheme. To optimize its computation, we introduce the *Multiplicative Jacobian Energy Decomposition* (MJED) method for discretizing hyperelastic materials on linear tetrahedral meshes which leads to faster matrix assembly than the standard Finite Element Method. Visco-hyperelasticity is modeled by Prony series while the mechanical effect of liver perfusion is represented with a linear Darcy law. Dynamic mechanical analysis has been performed on 60 porcine liver samples in order to identify some visco-elastic parameters. Finally, we show that liver deformation can be simulated in real-time on a coarse mesh and study the relative effects of the hyperelastic, viscous and porous components on the liver biomechanics.

Keywords: hyperelastic, viscous, porous, liver, real-time simulation, MJED.

1. Introduction

The simulation of soft tissue deformation has attracted a growing interest in the past 15 years both in the biomechanics and the medical image analysis communities. Indeed, medical image modalities such as MRI, CT or echography can describe with millimeter accuracy the anatomical shape of soft tissues but also their deformation through time series of images such as cine-MRI or electrocardiography. Modeling *in silico* the deformation of soft tissues is of high interest in particular for the following applications: surgical gesture training[1], therapy planning[2], physically-based image registration[3] as well as medical imaging diagnosis.

In this paper, we focus on the simulation of liver deformation in the context of surgery training. The overall objective is to build a simulator that can ease the training of young medical residents to perform some gestures specific to minimally invasive surgery such as laparoscopy, endoscopy, interventional radiology, etc. In such cases, it is required that soft tissue deformation be

simulated in real-time, *i.e.* at nearly 25 frames per second for the persistence of vision to take place. Obviously with such performance, the simulation algorithms used in surgical simulators can also be used for surgery planning or even computer animation. However, in surgery simulation, there are additional constraints of numerical stability during the occurrence of contact between soft tissue and (virtual) surgical instruments. Because those instruments are controlled by a human operator, large compression or extension of tissue can occur, sometimes leading to non-physical behavior such as inverted elements.

Because soft biological tissue behavior is rather complex and also not well characterized, its simulation in real-time is a very challenging task. The liver, for instance, is a porous material which typically undergoes large displacements, large deformations and is also strongly visco-elastic (more details in section 2.2). Early approaches for modeling the soft tissue behavior assumed a linear elastic behavior[4] discretized on finite elements which naturally leads to solving a linear system of equations whose inverse could eventually be precomputed[5].

Since linear elastic materials are not suited for large displacements, several authors in computer animation

URL: Stephanie.Marchesseau@inria.fr (Stéphanie Marchesseau)

have proposed corotational elastic models[6, 7] where linear elastic stiffness matrices are rotated for each element. Those corotational models however do not minimize a strain energy and therefore are not the discretization of a continuum formulation. Also, their behavior is very restricted to material linearity.

To simulate realistic soft tissue deformation, several authors have employed hyperelastic materials minimizing a continuum strain energy. For real-time computation, early approaches have been based on St. Venant Kirchhoff materials[1] which exhibit a linear stress-strain relationship. Significant speed-up can be obtained by using reduced basis of deformation[8] or by grouping expressions on edges, triangles and tetrahedra[9] when discretized on linear tetrahedra. Those approaches are however limited to this specific material which has the drawback of not behaving well under compression.

For general hyperelastic materials, authors have relied on the Finite Volume Method[10] to simulate soft-tissue deformation with explicit time integration schemes. Similarly, Miller *et al.*[11] have proposed the *Total Lagrangian Explicit Dynamic* (TLED) algorithm for which elastic forces are based on the reference configuration unlike the *Updated Lagrangian* method widely used in commercial FEM code. This approach has been combined with Prony series to model viscoelasticity and has been implemented on Graphics Processing Units (GPU)[12] to reach real-time computations. However the main limitation of this approach is that it relies on explicit time integration schemes which greatly simplifies the update at each time step but requires small time steps to keep the computation stable especially for stiff materials. Furthermore, with explicit schemes it is necessary to iterate multiple times to propagate applied forces from a node to the whole mesh.

In this article, we first introduce the *Multiplicative Jacobian Energy Decomposition* (MJED): a general algorithm to implement hyperelastic materials based on total Lagrangian FEM with implicit time integration schemes. An optimized approach to build the stiffness matrix is proposed which is used to solve a linear system of equations at each time step. Our algorithm allows some matrix precomputations to be performed thanks to a decomposition of the strain energy isolating the determinant of the deformation gradient J and the combination of shape vectors with fourth order elasticity tensors. Furthermore, a specific regularization of the stiffness matrix allows to cope with highly compressed elements.

A second contribution of this article is to propose a realistic biomechanical model of the liver which combines hyperelasticity, viscoelasticity as well as poroelasticity. The viscoelasticity of our liver model is based

on Prony series whose parameters have been experimentally estimated through a dynamic strain sweep testing. Furthermore, those parameters have been validated by reproducing *in silico* the experiments performed on liver samples. Finally, we take into account the porous medium of the liver parenchyma through a poro-elastic model which computes the fluid pressure and the resulting applied pressure on the solid phase.

2. Materials and Methods

The goal of this work is to construct a physically-realistic mechanical model of the liver that is suitable for the simulation of hepatic surgery. As such, the model should be as accurate as possible, but efficient enough to allow its application in real-time applications.

The first challenge is to characterize the mechanical behavior of the liver through rheology tests on liver tissue. For instance, Yamada[13] has estimated the liver behavior in compression with an elastic modulus varying from 7.8 to 42kPa. Dependence of hepatic tissue elastic modulus with frequency was proposed in 2002 by Ottensmeyer[14] and in 2005 by Valtorta *et al.*[15], contributing in this way to consider liver tissue as viscoelastic. More recently, this dependence between liver mechanical modulus and frequency was reinforced thanks to the development of *in vivo* imaging-based elastography systems. A comparison between *in vivo* ultrasound-based transient elastography and *in vitro* rheometry has been proposed in 2009 by Périchon *et al.*[16]. In Figure 1, we show how our frequency dependent mechanical measurements of liver tissues compares with the literature. For large displacements, the non-linear behavior of the liver was first analyzed by Liu and Bilston[17]: liver tissue was considered as non-linear beyond 0.2% strain. Kerdok [18] in her thesis has proposed a global mechanical model of the liver based on *ex-vivo* perfused porcine livers.

Based on this literature survey of liver biomechanics, we propose to model hepatic tissue as hyperelastic and viscous. Moreover, porous properties which are due to the large amount of extracellular fluid in the liver, as explained in [18], are considered here. Figure 2 shows schematically how these three components are related within a physically-based model. The porosity model acts in parallel to the visco-hyperelastic components and introduces the fluid pressure as an additional state variable. In the next sections, the individual components and underlying variables are described in details.

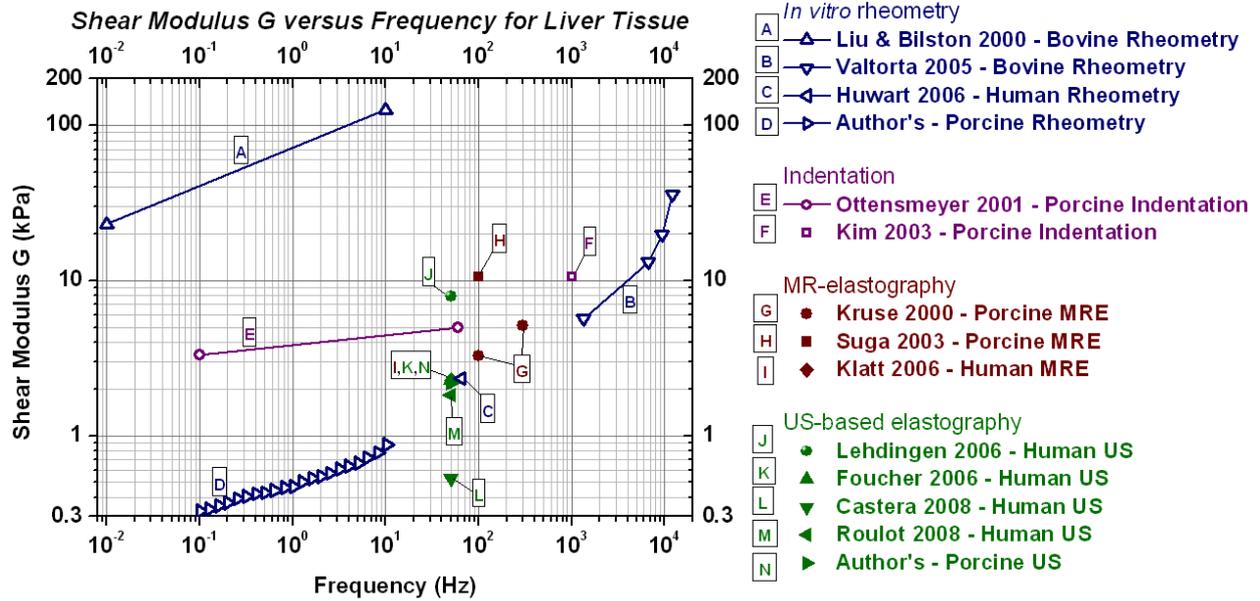


Figure 1: Comparison has been done between classical rheometry (blue), indentation (purple), Magnetic Resonance (red) and Ultrasound-based (green) Elastography tests.

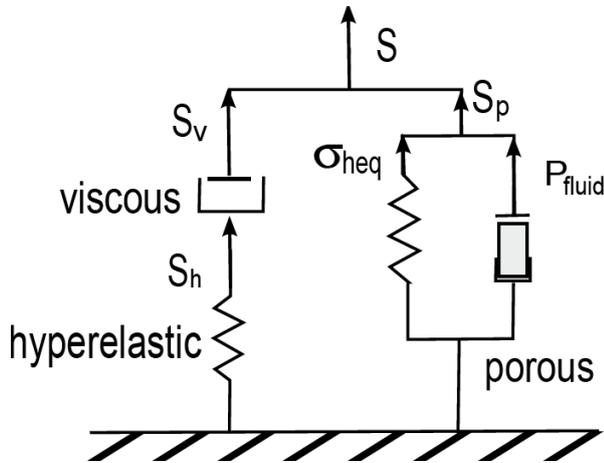


Figure 2: Representation of the constitutive model combining viscosity, hyperelasticity and porosity.

2.1. Fast computation of Hyperelastic Materials

Under large deformation, linear elasticity is no longer valid and the liver behavior is better represented as an hyperelastic material. Since we are using implicit time integration schemes, it is necessary at each time step to compute hyperelastic forces and stiffness matrices with a discretization method. The Finite Element Method (FEM) is a widely used approach to this end, however the constraint of real-time simulation is not always sat-

isfied. The objective of this section is to introduce a fast discretization method suitable for all hyperelastic materials and to compare it with classical FEM.

To discretize the liver geometry, we use tetrahedral linear finite elements because they are straightforward to generate from triangular surfaces that are outputted by image segmentation algorithms. Linear tetrahedral finite elements have some limitations since they can exhibit numerical locking when enforcing incompressibility[19]. However, as shown in section 2.3, the liver is not incompressible at the time scale considered due to the porous nature of the parenchyma. Note also that all optimizations developed in this section can be easily extended to other elements in particular linear hexahedral elements since it is based on the gradient of shape functions.

In tetrahedral finite elements, T_P is the rest tetrahedron (with vertices P_i) which is transformed under the deformation function $\phi(X)$ into the tetrahedron T_Q (with vertices Q_i). Any hyperelastic material is fully determined by its strain energy function W_h which describes the amount of energy necessary to deform the material. This strain energy function is defined in a way which is invariant to the application of rigid transformations: it involves the invariants of the *Cauchy-deformation tensor* defined as $\mathbf{C} = \nabla\phi^T\nabla\phi$. There are numerous invariants of \mathbf{C} (see [20] for detailed explanation) but the ones used for isotropic materials are the following: $I_1 = \text{tr}\mathbf{C}$,

$I_2 = \frac{1}{2}((tr\mathbf{C})^2 - tr\mathbf{C}^2)$ and the Jacobian $J = det\nabla\phi$. We define furthermore the deviatoric deformation tensor $\tilde{\mathbf{C}} = J^{-2/3}\mathbf{C}$, which by construction does not contain any volumetric dilation of the material but only pure deformation. Its first two invariants are written as \tilde{I}_1 and \tilde{I}_2 .

To capture the resistance to uniform compression or extension, a volumetric energy term $U(J)$ is added, as explained in section 2.3.

2.1.1. FEM Discretization of Hyperelastic Materials on Linear Tetrahedra

One way to consider finite elements method, for instance as explained by Zienkiewicz *et al.*[21], is to use the variational principle. The first variation of the strain energy is expressed as

$$\delta W = \int_{\Omega} \delta \mathbf{E}_{IJ} \mathbf{S}_{IJ} dV$$

where $\mathbf{S}_{IJ} = 2 \frac{\partial W}{\partial \mathbf{C}_{IJ}} = \frac{\partial W}{\partial \mathbf{E}_{IJ}}$ is the second Piola-Kirchhoff (SPK) stress tensor and \mathbf{E} is the Green-Lagrange strain tensor ($\mathbf{E} = 1/2(\mathbf{C} - \mathbf{Id})$). The SPK stress tensor and the Green-Lagrange strain tensor are energy conjugates of each other, similarly to force and displacement or torque and angle in other mechanical problems.

The discretization of those integrals on a tetrahedral mesh requires the definition of shape functions $\xi_i(X) \in \mathbf{R}$ for each point of tetrahedron T_p such that $\xi_i(P_j) = \delta_{ij}$, $\sum_i \xi_i(X) = 1$, and $\phi(X) = \sum_i \xi_i(X) Q_i$. On a linear tetrahedron, the shape functions $\xi_i(X)$ can be interpreted as the barycentric coordinate of X in T_p and can be written as : $\xi_i(X) = \mathbf{D}_i \cdot X + \alpha_i$ where \mathbf{D}_i is called a *shape vector*. Shape vectors $\mathbf{D}_i = \nabla \xi_i$ are the gradient of the shape functions and only depend on the geometry of the rest tetrahedron T_p . The four shape vectors \mathbf{D}_i in T_p have many remarkable properties among which they sum to zero (see definitions and properties in [22]).

Since the Green-Lagrange strain tensor is symmetric, it can be represented as a 6×1 vector. Its variation can then be computed as:

$$\delta \mathbf{E} = \sum_i \hat{\mathbf{B}}_i \delta \tilde{u}_i$$

where \tilde{u}_i is the nodal displacement and $\hat{\mathbf{B}}_i$ is the strain-displacement 6×3 matrix. The strain displacement matrix $\hat{\mathbf{B}}_i$ plays a major role in the classical formulation of FEM and its definition relies both on $\nabla\phi$ and \mathbf{D}_i . For instance, the nodal force \mathbf{F}_i at a vertex of tetrahedron T_p can be computed as :

$$\mathbf{F}_i(T_p) = - \int_{T_p} \hat{\mathbf{B}}_i^T \mathbf{S} dV$$

Similarly, the stiffness matrix \mathbf{K}_{ij} at an edge of tetrahedron T_p is formulated as :

$$\mathbf{K}_{ij} = \int_{T_p} \hat{\mathbf{B}}_i^T \hat{\mathbf{N}} \hat{\mathbf{B}}_j dV + \int_{T_p} \mathbf{D}_i^T \mathbf{S} \mathbf{D}_j dV$$

where $\hat{\mathbf{N}}$ is the 6×6 Voigt representation of the fourth order elasticity tensor given by $N_{IJKL} = 2 \frac{\partial S_{IJ}}{\partial C_{KL}}$.

This discretization method has the advantage of being general. However, it has three limitations. First, the formulation and the computation of the SPK stress tensor \mathbf{S} and the elasticity tensor $\hat{\mathbf{N}}$ can be fairly complex even for common hyperelastic materials such as Mooney-Rivlin for instance. One reason for this complexity is that the first and second derivatives of $J = det\nabla\phi$ with respect to \mathbf{C} are non trivial and involves the inversion of \mathbf{C} , e.g. $\frac{\partial J}{\partial \mathbf{C}} = \frac{1}{2} J \mathbf{C}^{-1}$. Since we use total Lagrangian method, this inversion needs to be done at each time step. This makes the expression of the derivatives of \tilde{I}_1 and \tilde{I}_2 particularly cumbersome and therefore computationally expensive to evaluate.

The second limitation is that the strain-displacement matrix $\hat{\mathbf{B}}_i$ combines two terms: the deformation gradient $\nabla\phi$ and the gradient of the shape functions \mathbf{D}_i . The former changes at each iteration while the latter is constant. For basic hyperelastic materials for which the elasticity tensor $\hat{\mathbf{N}}$ is constant (for instance St. Venant Kirchhoff materials), this is not optimal and a better choice would be to isolate the deformation gradient and to combine the shape vectors with the elasticity tensor.

The third limitation is that the classical FEM formulation of hyperelastic material cannot cope with flat ($J \equiv 0$) or even inverted tetrahedra ($J < 0$). Indeed, such deformation are non physical and do not represent a configuration managed in continuum mechanics. However, in interactive simulations, such cases of extreme compression can be met due to contact (penalty-based instead of being constraint based) with rigid objects or due to non-physical user-defined gestures. For instance, if the user-interface is not equipped with force-feedback, the tool controlled by the user can undergo non plausible trajectories. To improve robustness in these cases, it is important to propose a formulation for hyperelastic materials which can handle nearly flat tetrahedra.

2.1.2. Multiplicative Jacobian Energy Decomposition (MJED)

Our original discretization method addresses the three limitations of classical FEM. It consists of three separate contributions:

i) Decomposition of strain energy

Our approach is to decouple in the strain energy, the invariants of \mathbf{C} from J so as to avoid matrix inversions and complex derivative expressions. Instead of computing the force and stiffness matrix using the first and second derivative of the energy with respect to \mathbf{C} (leading respectively to \mathbf{S} and $\hat{\mathbf{N}}$), we compute them directly by deriving the energy with respect to the nodal position:

$$F_i = -\left(\frac{\partial W_h}{\partial Q_i}\right)^T \text{ and } \mathbf{K}_{ij} = \left(\frac{\partial^2 W_h}{\partial Q_j \partial Q_i}\right) \quad (1)$$

where W_h is the strain energy. It is important to note that the approach developed in this section is completely equivalent to the classical FEM one but leads to more efficient computation.

We propose to write the strain energy functions as a sum of terms

$$W_h^k = f^k(J)g^k(\tilde{I})$$

or its exponential, where $\tilde{I} = (I_1, I_2, I_4, \dots)$. Therefore g^k is independent of J , its derivative will not involve any matrix inversions. This decomposition applies to every studied case (Veronda Westmann, Arruda-Boyce, St. Venant Kirchhoff, NeoHookean, Ogden, Mooney Rivlin and the orthotropic Costa's law). That gives for instance

$$W_h = V_0 \sum_{k=1}^n f^k(J)g^k(\tilde{I}) + V_0 \exp\left(\sum_{k=n+1}^{n'} f^k(J)g^k(\tilde{I})\right) \quad (2)$$

Exponential terms here account for models such as Costa's or Veronda Westmann's (see [23] for a detailed description), the following calculations are made only for non exponential terms.

Using this decomposition of strain energy enables complex material formulation to be computed more efficiently with only a sum of reasonably simple terms and no matrix inversions. Once the decomposition is done, getting $f^{k'}(J)$ requires a 1D derivation, and getting $\mathbf{S}_h^k = 2\frac{\partial g^k(\tilde{I})}{\partial \mathbf{C}}$ requires to combine well-known derivatives of the invariants (such as $\frac{\partial I_1}{\partial \mathbf{C}} = \mathbf{Id}$ or $\frac{\partial I_2}{\partial \mathbf{C}} = \mathbf{Id}I_1 - \mathbf{C}$ where \mathbf{Id} is the 3×3 identity matrix). The full derivation, explained in Appendix A, gives

$$F_{h,i} = -V_0 \sum_{k=1}^n \left(f^{k'}(J)g^k(\tilde{I}) \left(\frac{\partial J}{\partial Q_i}\right)^T + f^k(J)\nabla\phi \mathbf{S}_h^k D_i \right)$$

where the derivative of the Jacobian is expressed as

$$\frac{\partial J}{\partial Q_i} = \frac{1}{6V_0}((Q_j - Q_i) \wedge (Q_k - Q_i))^T$$

ii) Formulation of the stiffness matrix

Implicit time integration schemes require the computation of the tangent stiffness matrix at each time step. This naturally involves elasticity tensors computed as the derivative of \mathbf{S}_h^k with respect to \mathbf{C} for each tetrahedron and at each time step. MJED leads to far simpler expressions of those tensors because \mathbf{S}_h^k is independent of J . Furthermore, in many common materials, we show that the term containing those elasticity tensors can be precomputed. The full expression of the stiffness matrix includes 6 terms that are detailed in Appendix A. We only focus below on the term involving the elasticity tensor:

$$\mathbf{R}^k = f^k(J) \left(\frac{\partial \mathbf{S}_h^k}{\partial Q_j} D_i\right)^T \nabla\phi^T$$

which requires the computation of the tensor $\frac{\partial \mathbf{S}_h^k}{\partial \mathbf{C}} : \mathbf{H}$ where \mathbf{H} is a symmetric matrix. In all cases, this tensor can be written as a sum of two kinds of terms,

$$\beta_1^k \mathbf{A}_1^k \mathbf{H} \mathbf{A}_1^k \text{ or } \beta_2^k (\mathbf{H} : \mathbf{A}_2^k) \mathbf{A}_2^k$$

where β_u^k are scalars, \mathbf{A}_u^k are symmetric matrices, and $\mathbf{A} : \mathbf{B} = \text{tr}(\mathbf{B}^T \mathbf{A})$ for any two matrices \mathbf{A}, \mathbf{B} . Therefore, the term \mathbf{R}^k is a combination of two terms:

$$f^k(J) \nabla\phi \mathbf{L}_1^k(i, j) \nabla\phi^T \text{ and } f^k(J) \nabla\phi \mathbf{L}_2^k(i, j) \nabla\phi^T$$

where $\mathbf{L}_1^k(i, j)$ and $\mathbf{L}_2^k(i, j)$ are linear matrices depending on the shape vectors D_i, D_j , the matrices \mathbf{A}_u^k and the scalars β_u^k .

This formulation leads to an optimization for the assembly of the stiffness matrix for two reasons. First, no fourth order tensors (often implemented as 6×6 matrices) are required, only scalars and symmetric matrices are involved in the computation. Second, except for the Ogden model, the matrices \mathbf{A}_u^k are constant and therefore matrices $\mathbf{L}_1^k(i, j)$ and $\mathbf{L}_2^k(i, j)$ can be precomputed for each tetrahedron before the simulation. Both features decrease the number of operations required (additions and multiplications).

iii) Coping with highly compressed elements

When tetrahedra are nearly flat, $1/J$ would normally tends to infinity. To avoid numerical instabilities while computing the force, the value of J was thresholded. However, the volumetric terms $f^k(J)$ in the strain energy still become dominant. This makes the stiffness matrix singular and thus leads to numerically unstable computations because there is an infinite number of deformed configurations leading to the same value of J . In order to cope with this, Teran *et al.*[24] perform

an SVD decomposition of the deformation gradient matrix. To avoid this computationally expensive decomposition, we propose instead to *regularize* the term $\mathbf{G}_h^k = f^{k''}(J) g^k(\tilde{I}) \frac{\partial J}{\partial Q_j} \otimes \frac{\partial J}{\partial Q_i}$ by replacing it with the following expression :

$$\mathbf{G}_h^k = f^{k''}(J) g^k(\tilde{I}) \left((1-h) \frac{\partial J}{\partial Q_j} \otimes \frac{\partial J}{\partial Q_i} + \frac{1}{3} h \frac{\partial J}{\partial Q_j} \cdot \frac{\partial J}{\partial Q_i} \mathbf{Id} \right)$$

The closer h is to 1, the closer the \mathbf{G}_h^k matrix is to a diagonal matrix. In practice, we set $h = (1 - J)$ if $0 \leq J \leq 1$, $h = 0$ if $J \geq 1$ and $h = 1$ if $J \leq 0$. In all cases, the trace of the regularized matrix is equal to the trace of the original matrix. By only regularizing the stiffness matrix, we still minimize the strain energy and therefore do not change the nature of the hyperelastic material. With this technique, it is even possible to handle inverted elements when the strain energy remains finite as $J = 0$.

2.2. Visco-hyperelasticity based on Prony series

To model accurately the viscoelasticity of the liver, Rayleigh damping cannot be used. Instead, we propose to rely on Prony series [12][25] which consists in adding to the hyperelastic stress tensor \mathbf{S}_v some time dependent stresses. This time dependence is given by $\alpha(t) = \alpha_\infty + \sum_i \alpha_i e^{-t/\tau_i}$ with the condition $(\alpha_\infty + \sum_i \alpha_i) = 1$. The visco-hyperelastic SPK tensor \mathbf{S}_v can be written as:

$$\mathbf{S}_v = \int_0^t \alpha(t-t') \frac{\partial \mathbf{S}_h}{\partial t'} dt' = \mathbf{S}_h - \sum_i \gamma_i$$

where $\gamma_i = \int_0^t \alpha_i (1 - e^{-(t-t')/\tau_i}) \frac{\partial \mathbf{S}_h}{\partial t'} dt'$

After a discretization over time this results in the recursive formula:

$$\gamma_i^n = a_i \mathbf{S}_h^n + b_i \gamma_i^{n-1} \text{ where } a_i = \frac{\Delta t \alpha_i}{\Delta t + \tau_i} \text{ and } b_i = \frac{\tau_i}{\Delta t + \tau_i}$$

Δt is the time step used for discretization and has to be the same as the time step for any solvers during the simulation.

The combination of the Prony series with our hyperelastic formulation only requires the computation of the inverse deformation gradient $\nabla \phi^{-1} = \left(\sum_{l=1}^4 P_l \otimes \frac{\partial J}{\partial Q_l} \right) / J$ (see Appendix B). Adding the viscous properties through the Prony series does not have a significant impact on the total computation times despite the evaluation of the time dependent stresses γ_i^n and $\nabla \phi^{-1}$.

2.3. Poro-elasticity

We propose to model the liver as a fluid-filled sponge following Kerdok's model [18], also described in [26].

The proportion of free-fluid (e.g. blood, water) in the liver parenchyma in the reference configuration is set to a constant f_w , $1 - f_w$ represents the initial ratio of the solid phase (hepatic cells). In such case, the volumetric part of the liver material (resistance to volume change) is governed by how much the volume change (measured by J) differs from $1 - f_w$. Thus, we introduce the effective volumetric Jacobian $J^* = (f_w + J - 1)/f_w$, and define the volumetric Cauchy stress following Hencky's elasticity [27]:

$$\sigma_{Heq} = K_0 f_w \ln(J^*)$$

where K_0 is the bulk modulus of the material. With this model, when J get close to $1 - f_w$, the solid phase of the liver is completely compressed and the resulting stress is infinite. To avoid instabilities due to this infinite stress, we substitute σ_{Heq} when $J \leq J_0$ by its tangent curve at J_0 (see Figure 3). We set $J_0 = 1 - f_w + K_0/K_{lim}$ where K_{lim} is a bulk modulus and represents the slope of the tangent.

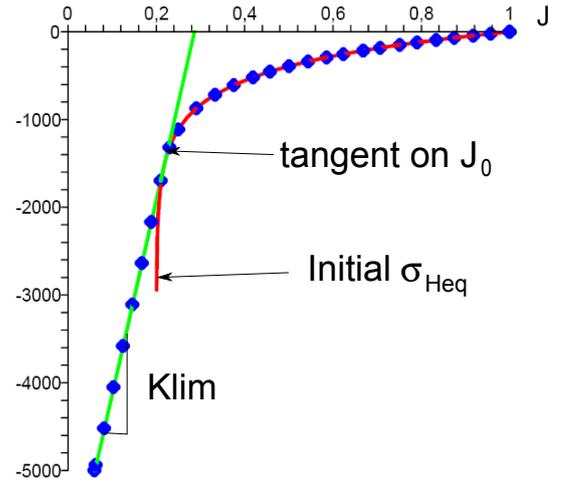


Figure 3: Representation of the static Cauchy stress before and after substitution. (Red) Initial stress, (Green) Tangent curve, (Blue dots) New stress. Here $f_w = 0.8$.

The fluid phase of the liver also applies some volumetric stresses due to the transient response of the fluid through the porous liver parenchyma. A straightforward way of modeling the porous behavior is through the linear Darcy's law. In this setting, variation of fluid pressure P_{fluid} is governed by the variation of volume change and a diffusive process:

$$\frac{1}{K_{lim}} \dot{P}_{fluid} = \kappa \nabla^2 P_{fluid} - \frac{\dot{J}}{J} \quad (3)$$

where κ is the permeability parameter. In Kerdok's

model, the permeability κ is a function of J , however its intensity varies at most of 10%. Therefore we propose to keep it constant to decrease its computational cost.

Finally, the total Cauchy stress response in the volumetric part is defined by summing the solid and the fluid terms: $\sigma_p = \sigma_{heq} \mathbf{Id} - P_{fluid} \mathbf{Id}$. The Cauchy stress is translated as a poro-elastic force and then added to the visco-hyperelastic forces:

$$F_{p,i} = -\sigma_p \left(\frac{\partial J}{\partial Q_i} \right)^T V_0$$

The additional stiffness matrices due to the porous behavior are given as:

$$\mathbf{K}_{p,ij} = V_0 \left(\frac{\partial \sigma_p}{\partial J} \left(\frac{\partial J}{\partial Q_j} \right)^T \left(\frac{\partial J}{\partial Q_i} \right) + \sigma_p \frac{\partial^2 J}{\partial Q_j \partial Q_i} \right)$$

The regularization strategy of the volumetric term described in section 2.1.2 iii) is applied to the stiffness matrix $\mathbf{K}_{p,ij}$.

The poro-elastic equation (3) is integrated with an Euler semi-implicit method at the same frequency and with the same time step as the mechanical equation. The diffusive part of Darcy's law is discretized implicitly as a constant stiffness matrix and at each time step a linear system of equation involving nodal fluid pressures is solved with a conjugated gradient method with a maximum number of iterations (to limit computational cost). Note however that the term \dot{J}/J discretized explicitly may undergo large variation and thus limits the maximal value of the global time step.

3. Results and Validation

3.1. Testing accuracy and computation time of the hyperelastic implementation

Decreasing computation time of the assembly of the stiffness matrix and force vector is essential to reach real-time simulation as this represents around 45% of the total time needed in one step. Therefore we first compared our implementation with the classical FEM method explained in [21], referred to as "Standard FEM", both implemented in SOFA¹. The results are given in Figure 4. We measured the time elapsed for the computation of the nodal forces and the stiffness matrices averaged over 100 iterations. We simulated the

deformation of a cube with 20 700 tetrahedra and 4300 nodes. For all modes implemented the proposed strategy is definitely more efficient than the standard FEM, up to five times as fast for St Venant Kirchhoff material.

Second, two sets of comparisons have been made to check the accuracy of the MJED. A first set of tests compares the node positions after a specified deformation between our simulation and the Open Source software FEBio (version 1.1.7)² where several elastic and hyper-elastic materials are implemented. The mean relative difference is around 10^{-6} for every models tested.

A second set of tests has been performed with the analytical solution of a deformed cube on which a vertical pressure p is applied on its top face. To this end, we assume that the deformation gradient in the global x-y-z coordinate system is $\nabla \phi = \text{diag}(\alpha, \beta, \gamma)$. The simulated values of α, β, γ are compared to analytical solutions of the system:

$$\begin{cases} e_z^T \nabla \phi \mathbf{S}_h e_z = p \\ e_x^T \nabla \phi \mathbf{S}_h e_x = 0 \\ e_y^T \nabla \phi \mathbf{S}_h e_y = 0 \end{cases}$$

Figure 5 shows the results in the non-linear domain, for St Venant Kirchhoff materials.

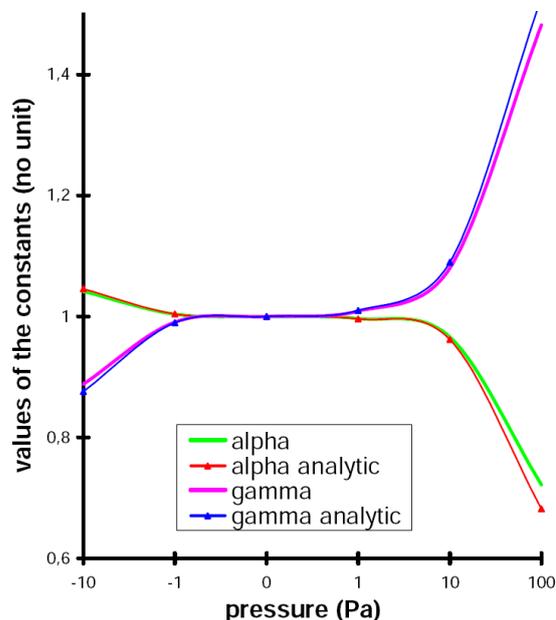


Figure 5: Analytical and computed results for α and γ for several pressures, with a log-scale, on St Venant Kirchhoff elasticity

¹SOFA is an Open Source medical simulation software available at www.sofa-framework.org

²FEBio is an opensource software package for FE analysis available at mrl.sci.utah.edu

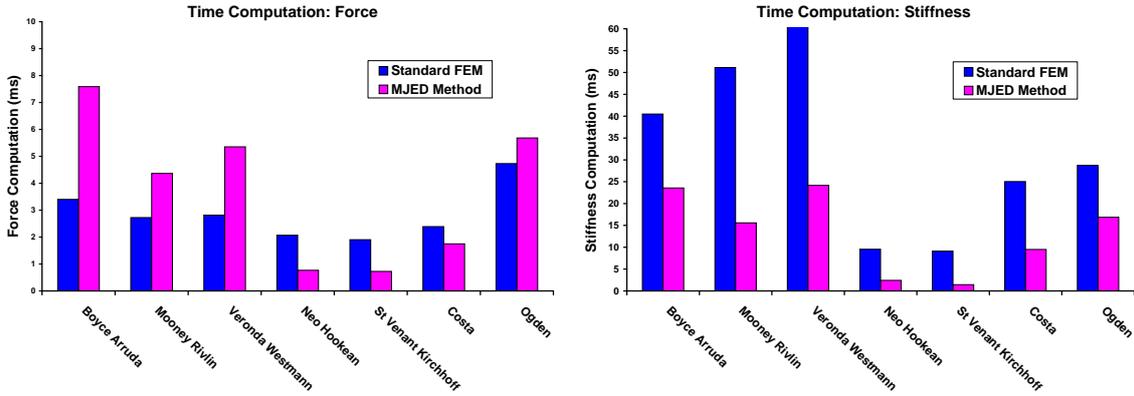


Figure 4: Comparison of the computation times of nodal forces and stiffness matrices between two different discretization methods averaged over 100 iterations.

It can be seen that the computed results are very close to the analytic tests, the mean difference for St Venant Kirchhoff on each constants is of order 10^{-4} in the linear domain and 10^{-2} for larger pressure. For Costa's law, the mean difference is of order 10^{-8} in the linear domain and 10^{-4} for larger pressure.

To conclude, with those two sets of tests, we have validated the accuracy of the MJED method for hyper-elastic materials. Moreover a mesh convergence study has been performed to evaluate the trade-off between accuracy and speed on several meshes. It appeared that multiplying the discretization of a mesh by 20 results in increasing the accuracy by 1% but dividing the number of frames per second by almost 10. Therefore a compromise has to be found to allow realistic interactions.

3.2. Small deformation dynamical testing on porcine liver: experimental materials and methods

To calibrate the visco-elastic parameters of our liver model, tests were performed on porcine livers. Dynamic viscoelastic behavior of hepatic tissue was investigated using *in vitro* Dynamic Mechanical Analysis (DMA) in rotating shear. Tests were carried out on liver cylindrical shaped samples coming from five adult pigs (weighting between 25 and 35kg). Immediately after hepatectomy performed on anesthetized animals, entire livers were stored in an insulated container at 6C surrounded by ice. Cylindrical hepatic samples of 20 mm diameter and 4 mm thick were cut and tested within 6 hours post-mortem time. To avoid mechanical difference due to samples localization, 4 samples were tested for each of right, middle and left liver lobe. Hepatic tissue is considered as isotropic. At least 60 samples were tested (5 animals x 3 lobes x 4 samples).

Dynamic Strain Sweep tests and Dynamic Frequency Sweep tests were performed on a dedicated stress-controlled AR2000 (TA-Instruments, New Castle, DE, USA) rheometer in a parallel-plate configuration represented in Figure 6. A pre-compression of 5mN was applied and sand paper was fixed to the rheometer plates to insure grip with tested sample.

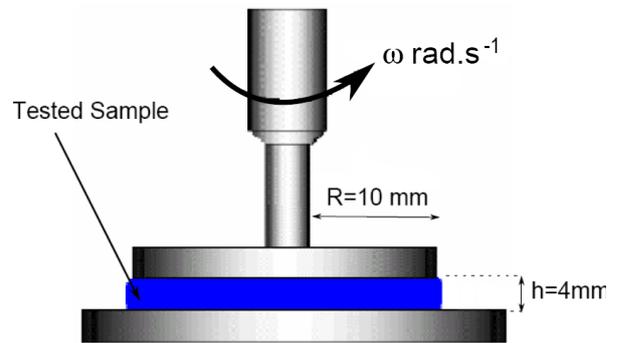


Figure 6: Schematic representation of the testing device in a plate-plate configuration. Lower plate is fixed whereas upper is sinusoidally rotating

- Dynamic Strain Sweep

These tests aim at determining the linear limit of the material elasticity. The sweep covers strain range from 0.01% to 20% which is sufficient according to the literature. Storage and loss shear moduli, noted respectively G' and G'' can be extracted from the experiments. In Figure 7 the storage modulus is given for two samples and at two different temperatures. The linear domain extends up to 1% to 2% of deformation.

- Dynamic Frequency Sweep

The upper plate applies a sinusoidal torque to the sample and causes a sinusoidal deformation shifted with a

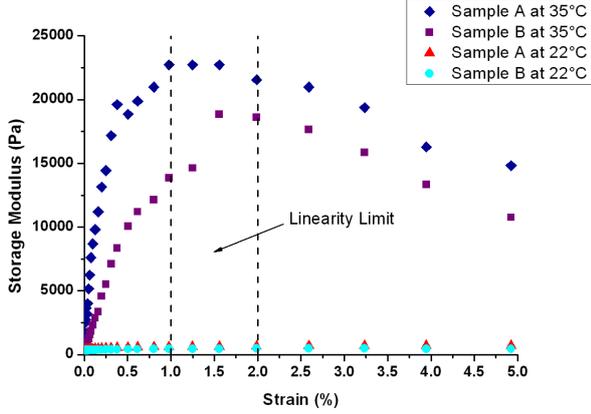


Figure 7: Strain Sweep tests: storage modulus for in vitro hepatic tissue samples.

phase δ . A frequency sweep from 0.1 to 4Hz can be generated. Experiments were carried out in the linear viscoelastic strain range of the samples ($\gamma_0 = 0.1\%$). Storage and loss shear moduli, as well as complex shear modulus G^* were calculated from the formulas:

$$G'(\omega) = \cos\delta \frac{\sigma_0}{\gamma_0}, G''(\omega) = \sin\delta \frac{\sigma_0}{\gamma_0} \quad (4)$$

and the complex modulus $G^*(\omega) = G'(\omega) + iG''(\omega)$ where σ_0 and γ_0 are the amplitude of the sinusoidally varying shear stress and strain respectively and δ the phase difference between stress and strain. We display the mean curves in Figure 8, obtained from the 60 tested samples.

From these results the Dynamic Modulus G can be obtained as a function of frequency or of time, and the viscoelastic behavior can be modeled after fitting a generalized Maxwell model with two modes of relaxation to those measurements:

$$G(t) = G_0(g_\infty + g_1 e^{-t/\tau_1} + g_2 e^{-t/\tau_2})$$

where $G_0 g_\infty = G_\infty$ is called the equilibrium modulus, g_1, g_2, τ_1, τ_2 are parameters such as $g_\infty + g_1 + g_2 = 1$. The parameters are given in Table 1. From Figure 1, we

$G_0(Pa)$	$g_1(Pa)$	$\tau_1(s)$	$g_2(Pa)$	$\tau_2(s)$
770	0.235	0.27	0.333	0.03

Table 1: Values of the parameters identified from the DMA Tests

can see that a huge disparity of results is reported from the literature. Each curve and value is a feature of specific loads (strain, strain rate) and experimental process (post-mortem time, species, location in the liver and cutting orientation of the samples) that takes more or less *in*

in vivo physiological parameters into account (blood pressure, vessels, temperature, hepatic tissue heterogeneity and anisotropy). Whereas *in vitro* experiments are not able to reproduce all the physiological conditions, *in vivo* tests are limited by more or less validated devices but seem to be more accurate for hepatic tissue characterization. At least, each test and associated curve is specific to one use of the resulting tissue model.

Temperatures of the samples have to be considered as well. While frequency sweep of shear modulus is not very dependent on temperature, it influences significantly strain sweep results for soft tissues. No assumption was done for liver tissue, however some authors showed on other biological soft tissues (human skin) that collagen fibers retract (from ambient temperature 22°C to 65°C) to the third of its initial length due to dislocations in the crystal structure. To avoid tissue spoiling, making tests at porcine temperature (35°C) would have required to be able to manage tissue moisturizing. Nevertheless, for frequency sweep DMA as well as for relaxation tests, the assumption of constant mechanical parameters from ambient temperature to 35°C was done in accordance with results proposed in the literature (soft tissue mechanical modifying only under 5°C and over 40°C).

Finally, our estimated shear modulus is significantly lower than those measured by other authors on human or bovine livers at the same frequency range. However, they are coherent with other measurements performed on porcine livers at higher frequencies with image-based elastography.

3.3. Model Parameter fitting from experiments

From the rheological experiments described in the previous section, we derive the shear modulus G_0 required in the hyperelastic term (e.g. Arruda-Boyce material) and the Prony series parameters required in the viscous term. To check that those parameters are indeed valid, we propose in this section to compare *in silico* simulations with the performed *in vitro* rheological tests. First, we check that the linear domain for our hyperelastic materials matches the ones observed in the dynamic strain sweep experiments. We simulate a cylinder in extension in SOFA for several longitudinal stresses and estimate the associated strains. The stress/strain curves are shown in Figure 8 for two materials. It can be seen that the linear limit corresponds to the one given by experiments (1%–2%). Second, dynamic frequency sweep tests have been simulated using similar geometries and boundary conditions to the DMA tests (see Figure 9). An oscillating torque (amplitude M) is applied on a small cylinder (radius

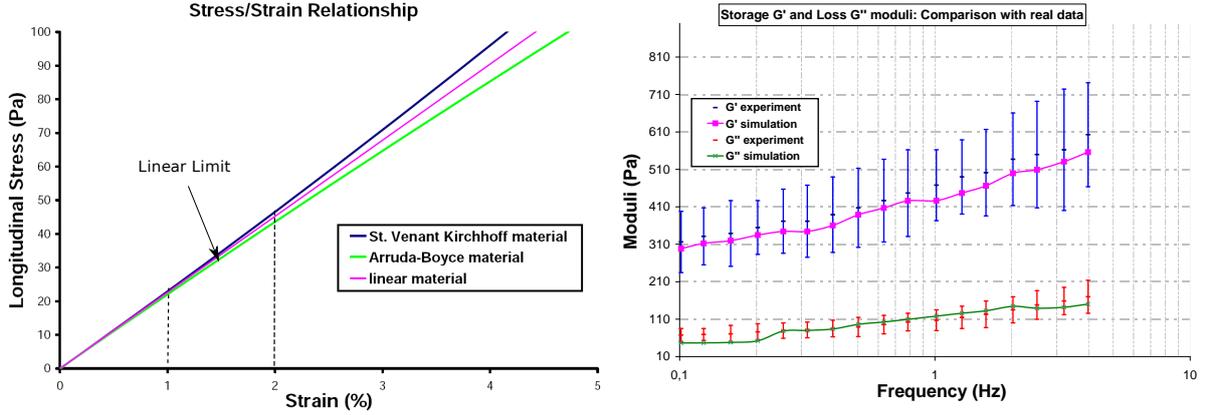


Figure 8: (Left) Longitudinal stress/strain curves obtained with a cylinder for Boyce Arruda and St Venant Kirchhoff materials. (Right) Comparison of the simulated values with the data obtained by DMA testing. The moduli are given on a x-log scale. The material is St Venant Kirchhoff, similar values are found for other materials.

$r = 10\text{mm}$, height $h = 4\text{mm}$) at various frequencies ω . The amplitude of the torque is chosen so as to stay in the linear domain. The angle of rotation θ of the cylinder is measured as a function of time. This angle describes a sinusoidal curve which follows the torque amplitude with a shifted phase δ . Specific constraint is applied on the top cylinder nodes to enforce a pure rotation of those nodes (as to reproduce the pure grip of sand paper).

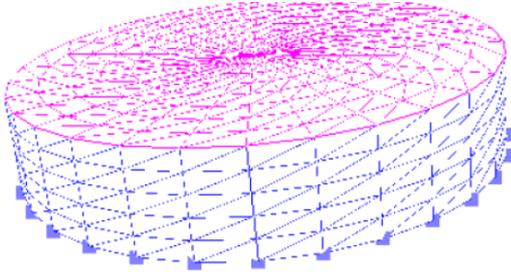


Figure 9: (Left) Cylinder used for the simulation of shear deformation. (Pink) Rotating top nodes. (Blue triangles) Cylinder with parameters of the liver. (Blue dots) Fixed points.

Applying the classical formulas:

$$\sigma(t) = \frac{2M}{\pi r^3} \text{ and } \gamma(t) = \frac{r \theta(t)}{h}$$

and equations (4) we can estimate the values of the storage and loss moduli to be compared with experimental data. It can be seen in Figure 8 that the simulation manages to capture the viscous behavior of the liver for small deformations with a mean relative error of 5%. Given the fit errors and the standard deviation of the values obtained with the DMA tests, this mean error is reasonably good.

3.4. Poro-hyperelasticity simulation

We have implemented the porous component in our model using parameters based on Kerdok's [18] experimental data as shown in Table 2.

f_w	$K_0(\text{Pa})$	$K_{lim}(\text{kPa})$	$\kappa(\text{m}^4/\text{Ns})$
0.5	400	2.2	20

Table 2: Values of the parameters used for porosity component

In order to qualitatively check the accuracy of our implementation, we simulated a liver composed of two components: Arruda-Boyce hyperelasticity and porosity. From time $t=0\text{s}$, gravity is uniformly applied to the liver while several selected vertices of a plane are fixed (representing the ligament and veins). The simulated fluid pressure field during the deformation is shown in Figure 10 as a color map, ranging from dark blue (initial pressure) to red (highest pressure). Highest pressure in the fluid occurs when the liver is compressed either by gravity (diffusion starts at the top) or by elastic reaction (diffusion starts at the bottom).

3.5. Complete Liver Model

To describe the influence of each component in the complete model, several simulations were performed using the same liver mesh (1240 vertices and 5000 tetrahedra), same parameters, under the same constant gravity force, and with a fifth-order Arruda-Boyce material [28]. The liver mesh has been segmented from a CT scan image and meshed with the GHS3D software. An Euler implicit time integration scheme is used with a time step of 0.07s, the linear equations being solved with a conjugated gradient algorithm. As

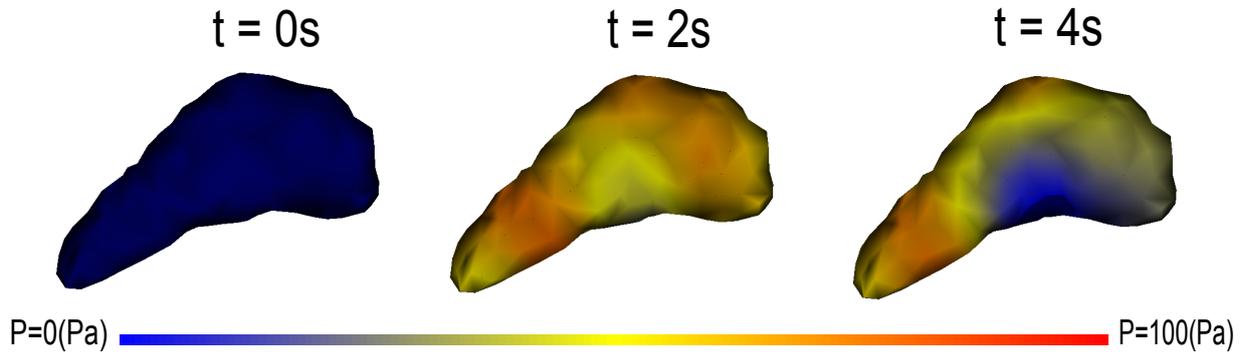


Figure 10: Pressure field of the porous component on a liver under gravity

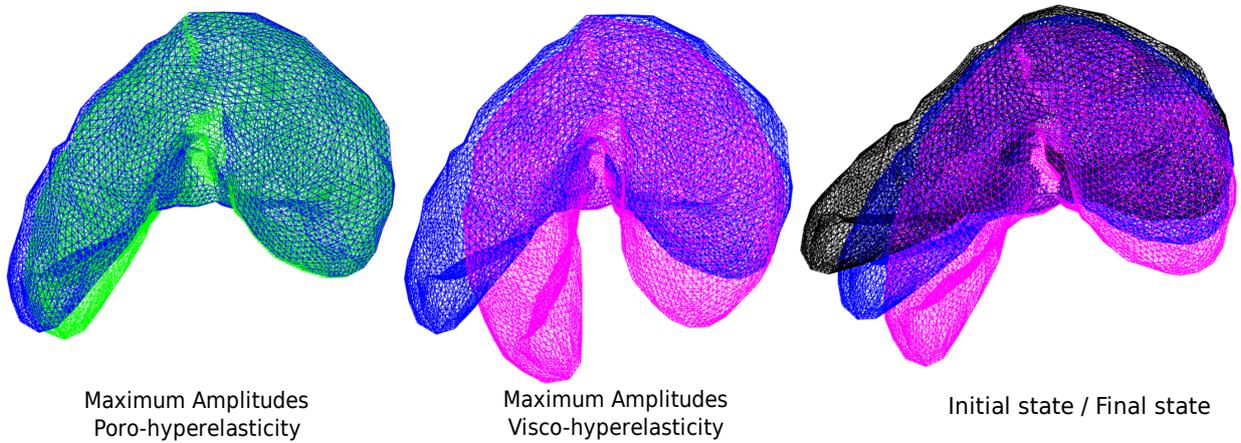


Figure 11: Addition of viscosity or porosity to hyperelasticity: Comparison of the maximum amplitudes and final states. (Black) Initial position, (Blue) Hyperelastic liver, (Pink) Visco-hyperelastic liver, (Green) poro-hyperelastic liver.

boundary conditions, several nodes of the liver are fixed along the vena cava and suspensive ligament. The liver deforms under the action of gravity exceeding the linear limit of the material. All computations were performed on a laptop PC with a Intel Core Duo processor at 2.80 GHz (simulations are available in the video clip). The liver motion could be described as a pendulum-like motion around the equilibrium state until stabilization.

(i) Influence of the viscous component

Adding viscosity to hyperelasticity increases the amplitude of the oscillations as the material becomes less stiff. In contrast to an essentially hyperelastic model, the final state is much different from the initial state (see Figure 11). Indeed, the use of Prony series leads to a multiplication of the SPK tensor by $1 - \sum a_k$ at infinite time, which decreases the resistance of the liver. The frame rate is around 13 FPS against 14 FPS for hyperelasticity alone. We did not reach the ideal 25 FPS needed for real-time interaction. However the

implicit integration scheme allows larger time step (0.3s for instance) which speeds up the simulation and makes user interactions efficient. High amount of extension and compression are possible which may be somewhat unrealistic, therefore the porous component is necessary to control the amount of viscosity.

(ii) Influence of the porous component

Adding porosity to visco-hyperelasticity prevents the liver from having unrealistic large deformations. The maximum amplitude is in between the hyperelasticity alone and the visco-hyperelasticity. The deformation is no longer isotropic and changes over time. The addition of this component decreases the computational efficiency (6 FPS) since a semi-implicit integration scheme is used for the porous component. Because of the fast variation of the explicit term \dot{J}/J , the time step has to be decreased to 0.15s. On our laptop PC, the simulation is still fluid enough to allow user interactions.

4. Discussion

In this paper we have proposed an innovative method to discretize hyperelastic materials on finite element meshes. The MJED method is fully general and requires the user to provide a decomposition of the strain energy into simple terms. With this formulation, a number of precomputations can be performed to speed-up the assembly of stiffness matrices.

For the complete liver model, Arruda-Boyce material was chosen based on Kerdok's observations [18] that the parenchyma is best represented by a 8-chain rubber like elastic model. But alternative materials such as Mooney Rivlin or Ogden could be also employed, whereas a Costa material is better suited for cardiac modeling. In the future, we are planning to perform additional strain sweep static tests in order to better characterize the hyperelastic behavior of the liver. One improvement of our model would be to add the influence of the Glisson capsule which acts as a membrane surrounding the liver parenchyma. However rheological experiments of that capsule are difficult to perform because it is very thin, other type of experiments such as tensile testing could be considered. Another avenue of research would be to couple the liver perfusion model with the simulation of blood flow inside the two liver veinous systems.

We have also shown in section 3.3 that some parameters of the liver model can be identified based on dynamic mechanical testing. Model personalization is an important issue to create patient-specific simulations and we believe that ultrasound or MR elastometry could be used for *in vivo* characterization of liver visco-elasticity. However such experiments are only valid for small deformations since they are non-invasive.

For full user interaction, it is required to reach at least 25 FPS. For our visco-hyperelastic liver model, we were still able to have a reasonable interaction despite a frame rate of 13 by increasing the time step to 0.3s. Note that using implicit time integration schemes helped to obtain realistic behavior, whereas using explicit schemes with much lower time steps and large frame rate lead to very damped motion. For instance, the same hyperelastic model implemented with an explicit Runge Kutta 4 solver could not be stabilized after the first interaction even for a small time step (0.01s), and reaching only 11 FPS. Thus, to have a robust simulation, no explicit solver could be chosen.

There are several ways to improve the computational efficiency (besides applying Moore's law). First, the porosity computation could be computed in parallel with the mechanical computation. Second, the assembly of stiffness matrices and the solution of linear sys-

tems of equations could be done on the GPU as already demonstrated by several authors[12]. Finally, the projection on reduced basis as shown in [8] could decrease the size of the linear system to be solved at each time step.

5. Conclusion

In this paper, we have introduced an efficient method to assemble stiffness matrices for complex biomechanical material which compares favorably with the standard FEM method. We have also proposed a poro-visco-hyperelastic liver model suitable for real-time interaction which is, up to our knowledge, among the most realistic ones in the literature. Several model parameters have been identified from rheometric tests performed on 60 samples from porcine livers and a validation study has been conducted to reproduce those tests well. Finally, the influence of each mechanical component has been evaluated. Despite those advances, much research needs to be done to achieve a realistic liver surgery simulation including the modeling of liver contact with neighboring structures, the influence of breathing and cardiac motion, the simulation of hepatic resection, bleeding and suturing.

Appendix A. MJED method

The strain energies $W_h^k = f^k(J)g^k(\tilde{I})$ are derived with respect to the points Q_i to obtain nodal forces, (described previously in equations 1) and the terms are summed up for each k . To get this first derivative, we use the same calculations as made by Delingette [22].

$$\text{Combining } \nabla\phi = \sum_{i=1}^4 Q_i D_i^T \text{ and } \mathbf{S}_h^k = 2 \frac{\partial g^k(\tilde{I})}{\partial \mathbf{C}}$$

we obtain:

$$\frac{\partial g^k(\tilde{I})}{\partial Q_i} = D_i^T \mathbf{S}_h^k \nabla\phi^T \quad \text{and from} \quad \frac{\partial f^k(J)}{\partial Q_i} = f^{k'}(J) \frac{\partial J}{\partial Q_i} \quad (\text{A.1})$$

where the derivative of the Jacobian is expressed as

$$\frac{\partial J}{\partial Q_i} = \frac{1}{6V_0} ((Q_j - Q_i) \wedge (Q_k - Q_i))^T \quad (\text{A.2})$$

we get nodal forces that only require the inputs $f^k, f^{k'}, g^k$ and \mathbf{S}_h^k :

$$F_{h,i} = -V_0 \sum_{k=1}^n \left(f^{k'}(J) g^k(\tilde{I}) \left(\frac{\partial J}{\partial Q_i} \right)^T + f^k(J) \nabla\phi \mathbf{S}_h^k D_i \right) \quad (\text{A.3})$$

To obtain the stiffness matrix, we need to derive twice the strain energy, or to derive the transpose of the force. We start by deriving the first term of the force: $f^{k'}(J) \frac{\partial J}{\partial Q_i} g^k(\tilde{I})$. We obtain three terms:

$$\begin{cases} \mathbf{G}^k = \left(\frac{\partial f^{k'}(J)}{\partial Q_j} \right)^T \frac{\partial J}{\partial Q_i} g^k(\tilde{I}) \\ \mathbf{H}^k = f^{k'}(J) \frac{\partial^2 J}{\partial Q_j \partial Q_i} g^k(\tilde{I}) \\ \mathbf{I}^k = f^{k'}(J) \left(\frac{\partial g^k(\tilde{I})}{\partial Q_j} \right)^T \frac{\partial J}{\partial Q_i} \end{cases} \quad (\text{A.4})$$

which are easily written using equation (A.1) and the second derivative of the Jacobian:

$$\frac{\partial^2 J}{\partial Q_j \partial Q_i} = \frac{1 - \delta_{ij}}{6V_0} \begin{bmatrix} 0 & -c_3 & c_2 \\ c_3 & 0 & -c_1 \\ -c_2 & c_1 & 0 \end{bmatrix} \quad (\text{A.5})$$

with δ_{ij} the Kronecker delta and $\mathbf{c} = (c_1, c_2, c_3)$ the edge vector opposing \mathbf{Q}_i and \mathbf{Q}_j .

Let consider the second term of the force: $f^k(J) D_i^T \mathbf{S}_h^k \nabla \phi^T$. As D_i is constant, that also leads to three additional terms:

$$\begin{cases} \mathbf{\Lambda}^k = \left(\frac{\partial f^k(J)}{\partial Q_j} \right)^T D_i^T \mathbf{S}_h^k \nabla \phi^T \\ \mathbf{M}^k = f^k(J) D_i^T \mathbf{S}_h^k \left(\frac{\partial \nabla \phi^T}{\partial Q_j} \right)^T \\ \mathbf{R}^k = f^k(J) \left(\frac{\partial \mathbf{S}_h^k}{\partial Q_j} D_i \right)^T \nabla \phi^T \end{cases} \quad (\text{A.6})$$

Computation of $\mathbf{\Lambda}^k$ and \mathbf{M}^k is straightforward using the definition of $\nabla \phi$ which gives $\frac{\partial \nabla \phi}{\partial Q_j} = \mathbf{I} \mathbf{d} D_j^T$ and equation (A.1). \mathbf{R}^k is more complex and in the remainder the \cdot^k notation is dropped to simplify notations and derive the expression component wise. We seek to determine a matrix whose elements are $\frac{\partial \mathbf{S}_{ab}}{\partial Q_j^v}$ where a, b and v are in [1..3]. Including the matrix \mathbf{C} in the chain rule we express this term as:

$$\frac{\partial \mathbf{S}_{ab}}{\partial Q_j^v} = \sum_{m,n=1}^3 \frac{\partial \mathbf{S}_{ab}}{\partial \mathbf{C}_{mn}} \frac{\partial \mathbf{C}_{mn}}{\partial Q_j^v}$$

But, using $\frac{\partial \mathbf{C}_{mn}}{\partial Q_j^v} = \sum_{u=1}^4 [Q_u^v D_j^m D_u^n + Q_u^v D_u^m D_j^n]$ and $\nabla \phi_{mn} = \sum_{u=1}^4 Q_u^m D_u^n$ and noticing moreover that $\mathbf{D}_j^m \nabla \phi_{vn} = [D_j \otimes (\nabla \phi^T e_v)]_{mm}$. We can finally compute the expressions to obtain the derivative with respect to \mathbf{Q}_j^v :

$$\frac{\partial \mathbf{S}}{\partial Q_j^v} = \frac{\partial \mathbf{S}}{\partial \mathbf{C}} : [D_j \otimes (\nabla \phi^T e_v) + (\nabla \phi^T e_v) \otimes D_j]$$

where $\frac{\partial \mathbf{S}}{\partial \mathbf{C}}$ is a fourth order tensor, applied to the matrix $[D_j \otimes (\nabla \phi^T e_v) + (\nabla \phi^T e_v) \otimes D_j]$ which is a second order

tensor (a matrix). Finally the derivative we seek is :

$$\frac{\partial \mathbf{S}}{\partial Q_j} D_i = \sum_{v=1}^3 \frac{\partial \mathbf{S}}{\partial Q_j^v} D_i \otimes e_v$$

Furthermore, we take advantage of the specific structure of the fourth order elasticity tensors. In all cases, this tensor can be written as a sum of two kinds of terms,

$$\beta_1^k \mathbf{A}_1^k \mathbf{H} \mathbf{A}_1^k \text{ or } \beta_2^k (\mathbf{H} : \mathbf{A}_2^k) \mathbf{A}_2^k$$

where β_u^k are scalars, \mathbf{A}_u^k are symmetric matrices, and $\mathbf{A} : \mathbf{B} = \text{tr}(\mathbf{B}^T \mathbf{A})$ for any two matrices \mathbf{A}, \mathbf{B} . Each one of those two kinds of terms leads to simpler expressions of \mathbf{R}^k , respectively:

$$f^k(J) \nabla \phi \mathbf{L}_1^k(i, j) \nabla \phi^T \text{ and } f^k(J) \nabla \phi \mathbf{L}_2^k(i, j) \nabla \phi^T \quad (\text{A.7})$$

where $\mathbf{L}_1^k(i, j)$ and $\mathbf{L}_2^k(i, j)$ are the linear matrices

$$\begin{cases} \mathbf{L}_1^k(i, j) = \beta_1^k (\mathbf{A}_1^k D_i \otimes D_j \mathbf{A}_1^k + \mathbf{A}_1^k (D_j \cdot \mathbf{A}_1^k D_i)) \\ \mathbf{L}_2^k(i, j) = 2\beta_2^k (\mathbf{A}_2^k D_j \otimes D_i \mathbf{A}_2^k) \end{cases} \quad (\text{A.8})$$

which are constant in most cases.

To conclude, the stiffness matrix is:

$$\mathbf{K}_{h,ij} = \frac{\partial^2 W(T_P)}{\partial Q_i \partial Q_j} = V_0 \sum_k (\mathbf{G}^k + \mathbf{H}^k + \mathbf{I}^k + \mathbf{\Lambda}^k + \mathbf{M}^k + \mathbf{R}^k) \quad (\text{A.9})$$

Appendix B. Combining MJED and Prony series

It consists in adding to the hyperelastic stress tensor \mathbf{S}_v some time dependent stresses:

$$\alpha(t) = \alpha_\infty + \sum_i \alpha_i e^{-t/\tau_i} \text{ with } \left(\alpha_\infty + \sum_i \alpha_i \right) = 1$$

At time n the visco-hyperelastic SPK tensor \mathbf{S}_v^n can be written as:

$$\mathbf{S}_v^n = \mathbf{S}_h^n - \sum_i \gamma_i^n$$

$$\gamma_i^n = a_i \mathbf{S}_h^n + b_i \gamma_i^{n-1} \text{ where } a_i = \frac{\Delta t \alpha_i}{\Delta t + \tau_i} \text{ and } b_i = \frac{\tau_i}{\Delta t + \tau_i}$$

Δt is the time step used for discretization and has to be the same as the time step for any solvers during the simulation. Therefore, we need to compute the total second Piola-Kirchoff stress tensor \mathbf{S}_h^n . This is done by computing the inverse deformation gradient :

$$\mathbf{S}_h^n = \nabla \phi^{-1} \left(\sum_k (f^{k'}(J) g^k(\tilde{I}) J \nabla \phi^{-T} + f^k(J) \nabla \phi \mathbf{S}_h^k) \right)$$

where $\nabla\phi^{-1} = \left(\sum_{l=1}^4 P_l \otimes \frac{\partial J}{\partial Q_l}\right) / J$.

The visco-hyperelastic nodal forces are therefore related to the hyperelastic ones by

$$F_{v,i}^n = F_{h,i}^n + V_0 \nabla\phi \sum_i \gamma_i^n D_i$$

Moreover, once we have γ_i^{n-1} the stiffness matrix is also slightly updated from its hyperelastic formulation:

$$\mathbf{K}_{v,ij}^n = \mathbf{K}_{h,ij}^n \left(1 - \sum_k a_k\right) - V_0 D_j^T \left(\sum_k b_k \gamma_k^{n-1}\right) D_i \mathbf{Id}$$

Appendix C. pseudo-code of the MJED method

To compute the FEM formulation, three functions are necessary to input into the solvers: an initialization, the computation of the force vector and finally the computation of the stiffness matrix. We start with the `init()` function (Algorithm 1) which loads the mesh, initializes the variables and precomputes some quantities. This function is only called once, before the simulation starts. Then we build the Force Vector (Algorithm 2) for every tetrahedron, and at each time step. And finally we compute the stiffness matrices (Algorithm 3 for every tetrahedron, at each time step (depending on the solver used).

Algorithm 1 `init()` function

- 1: Load the mesh and get the topology (tetrahedra, vertices, edges ...)
 - 2: **for** every tetrahedron `i` **do**
 - 3: Get the reference points position in the tetrahedron, P_j
 - 4: Calculate the reference volume V_0
 - 5: Calculate the Shape Vectors D_j
 - 6: **for** every edge with vertices `u` and `v` **do**
 - 7: Compute the linear matrices $\mathbf{L}_k(u, v)$ (see equation A.8)
 - 8: **end for**
 - 9: **end for**
-

Acknowledgements

This work is partially funded by the European PASS-PORT project (Patient-Specific Simulation for Pre-Operative Realistic Training of Liver Surgery) FP7-ICT-2007- 223894

[1] H. Delingette, N. Ayache, Soft tissue modeling for surgery simulation, in: Computational Models for the Human Body, Elsevier, 2004, pp. 453–550.

Algorithm 2 `addForce()` function

- 1: **for** every tetrahedron `i` **do**
 - 2: Get the current points position in the tetrahedron, Q_j
 - 3: Compute the Deformation Gradient $\nabla\Phi$
 - 4: Compute the Cauchy-Green Tensor \mathbf{C}
 - 5: Compute the Jacobian J
 - 6: **for** every vertex `j` **do**
 - 7: Compute the first derivative of the Jacobian $\frac{\partial J}{\partial Q_j}$ (see equation A.2)
 - 8: **end for**
 - 9: **for** every `k` of the decomposition **do**
 - 10: Get f_k, f'_k, f''_k, g_k , and \mathbf{S}_k
 - 11: **end for**
 - 12: **for** every vertex `j` **do**
 - 13: Compute the Force F_j summing the contributions from the decomposition (see equation A.3)
 - 14: **end for**
 - 15: **end for**
-

Algorithm 3 `addDForce()` function

- 1: **for** every tetrahedron `i` **do**
 - 2: **for** every edge with vertices `u` and `v` **do**
 - 3: Compute the second derivative of the Jacobian $\frac{\partial^2 J}{\partial Q_u \partial Q_v}$ (see equation A.5)
 - 4: **for** every `k` of the decomposition **do**
 - 5: Calculate $\mathbf{G}_k, \mathbf{H}_k, \mathbf{I}_k, \mathbf{\Lambda}_k, \mathbf{M}_k$ and \mathbf{R}_k using $\mathbf{L}_k(u, v)$ (equations A.4, A.6 and A.7)
 - 6: **end for**
 - 7: Compute the Stiffness Matrix \mathbf{K}_{uv} (see equation A.9)
 - 8: **end for**
 - 9: **end for**
-

- [2] D. Hawkes, D. Barratt, J. Blackall, C. Chan, P. Edwards, K. Rhode, G. Penney, J. McClelland, D. Hill, Tissue deformation and shape models in image-guided interventions: a discussion paper, *Medical Image Analysis* 9 (2) (2005) 163–175.
- [3] A. I. Veress, G. T. Gullberg, J. A. Weiss, Measurement of strain in the left ventricle during diastole with cine-mri and deformable image registration, *Journal of Biomechanical Engineering* 127 (7) (2005) 1195–1207.
- [4] S. Cotin, H. Delingette, N. Ayache, A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation, *The Visual Computer* 16 (8) (2000) 437–452.
- [5] D. L. James, D. K. Pai, ArtDefo accurate real time deformable objects, in: *Computer Graphics (SIGGRAPH)*, 1999, pp. 65–72.
- [6] M. Müller, J. Dorsey, L. McMillan, R. Jagnow, B. Cutler, Stable real-time deformations, in: *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2002, pp. 49–54.
- [7] M. Nesme, Y. Payan, F. Faure, Efficient, physically plausible finite elements, in: *Eurographics 2005*, August, 2005, Trinity College, Dublin, Ireland, 2005.
- [8] J. Barbič, D. L. James, Real-time subspace integration for St. Venant-Kirchhoff deformable models, *ACM TOG (SIGGRAPH)* 2005 24 (3) (2005) 982–990.
- [9] G. Picinbono, H. Delingette, N. Ayache, Non-Linear Anisotropic Elasticity for Real-Time Surgery Simulation, *Graphical Models* 65 (5) (2003) 305–321.
- [10] J. Teran, S. Blemker, V. N. T. Hing, R. Fedkiw, Finite volume methods for the simulation of skeletal muscle, in: *Eurographics/SIGGRAPH Symposium on Computer Animation*, San Diego, California, 2003, pp. 68–74.
- [11] K. Miller, G. Joldes, D. Lance, A. Wittek, Total lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation, *Communications in Numerical Methods in Engineering* 23 (2006) 121–134.
- [12] Z. Taylor, O. Comas, M. Cheng, J. Passenger, D. Hawkes, D. Atkinson, S. Ourselin, On modelling of anisotropic viscoelasticity for soft tissue simulation: numerical solution and gpu execution, *Medical Image Analysis* 13 (2009) 234–244.
- [13] H. Yamada, *Strength of Biological Materials*, Williams and Wilkins Co, 1970.
- [14] M. Ottensmeyer, Tempest 1-d: an instrument for measuring solid organ soft tissue properties, *Experimental Techniques* 26 (3) (2002) 48–50.
- [15] D. Valtorta, E. Mazza, Dynamic measurements of soft tissue viscoelastic properties with a torsional resonator device, *Medical Image Analysis* 9 (5) (2005) 481–490.
- [16] N. P. J. Oudry, S. Chatelin, L. Sandrin, A. Allemann, L. Soler, R. Willinger, In vivo liver tissue mechanical properties by transient elastography: Comparison with dynamic mechanical analysis, *International Research Council on Biomechanics of Injury* 2009.
- [17] Z. Liu, L. Bilston, Large deformations shear properties of liver tissue, *Biorheology* 39 (6) (2002) 735–742.
- [18] A. E. Kerdok, Characterizing the nonlinear mechanical response of liver to surgical manipulation, Ph.D. thesis, Harvard University (2006).
- [19] G. Joldes, A. Wittekemail, K. Miller, Suite of finite element algorithms for accurate computation of soft tissue deformation for surgical simulation, *Medical Image Analysis* 13 (6) (2009) 912–919.
- [20] J. Weiss, B. Maker, S. Govindjee, Finite element implementation of incompressible, transversely isotropic hyperelasticity, *Computer Methods in Applied Mechanics and Engineering* 135 (1996) 107–128.
- [21] C. Zienkiewicz, R. Taylor, *The Finite Element Method, Volume 2 : Solid Mechanics*, Butterworth-Heinemann, 2000.
- [22] H. Delingette, Triangular springs for modeling nonlinear membranes, *IEEE Transactions on Visualization and Computer Graphics* 14 (2) (2008) 329–341.
- [23] R. W. D. Veronda, Mechanical characterization of skin finite deformations, *Journal of Biomechanics* 3 (1) (1970) 111–124.
- [24] J. Teran, E. Sifakis, S. S. Blemker, V. Ng-Thow-Hing, C. Lau, R. Fedkiw, Creating and simulating skeletal muscle from the visible human data set, *IEEE Transactions on Visualization and Computer Graphics* 11 (3) (2005) 317–328.
- [25] J. Kim, M. A. Srinivasan, Characterization of viscoelastic soft tissue properties from in vivo animal experiments and inverse fe parameter estimation, in: *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, LNCS, Palm Springs, USA, 2005.
- [26] S. Raghunathan, D. Evans, J. L. Sparks, Poroviscoelastic modeling of liver biomechanical response in unconfined compression, *Annals of Biomechanical Engineering* 38 (5) (2010) 1789–1800.
- [27] H. Xiao, L. Chen, Hencky's elasticity model and linear stress-strain relations in isotropic finite hyperelasticity, *Acta Mechanica* 157 (2002) 51–60.
- [28] E. Arruda, M. Boyce, A three-dimensional constitutive model for the large stretch behavior of rubber elastic materials, *J. Mech. Phys. Solids* 41 (2) (1993) 389–412.